Constrained Trajectory Optimization for Kinematically Redundant Arms

Craig R. Carignan

Janice M. Tarrant

To appear in the ISRAM '90
Third International Symposium
on Robotics and Manufacturing,
July 18-20, 1990
Vancouver, B.C., Canada

CONSTRAINED TRAJECTORY OPTIMIZATION FOR KINEMATICALLY REDUNDANT ARMS

Craig R. Carignan and Janice M. Tarrant
ST Systems Corp.
Lanham, MD 20706

ABSTRACT

Two velocity optimization schemes for resolving redundant joint configurations are compared. The Extended Moore-Penrose Technique minimizes the joint velocities and avoids obstacles indirectly by adjoining a cost gradient to the solution. A new method can incorporate inequality constraints directly to avoid obstacles and singularities in the workspace. A four-link arm example is used to illustrate singularity avoidance while tracking desired end-effector paths.

1 INTRODUCTION

Kinematic redundancy can be defined as having more degrees of freedom than necessary to perform a particular task. For a three-dimensional task, the maximum number of end-effector positions which can be specified is six, three positions and three orientations. Thus a manipulator with more than six degrees of freedom is generally considered "redundant." Though redundancy imposes greater burden during trajectory planning, it is often a desirable property from the viewpoint of avoiding obstacles and gaining workspace accessibility.

Approaches to resolving the excess degrees of freedom in a kinematically redundant arm typically fall into one of two categories: (a) specify an additional number of equality constraints equal to the degree of arm redundancy, or (b) optimize a cost which reflects some desirable local property of the manipulator. The former case is called the "Extended Jacobian" technique [1] because the Jacobian is extended until it becomes a square 1-to-1 mapping between the extended end-effector velocity space and joint velocity space. The latter case is often called the "generalized inverse" technique because the resulting solution involves a weighted pseudoinverse of the Jacobian.

This paper focuses on local velocity optimization approaches for solving the redundancy problem. First, the unconstrained problem is solved resulting in the well known Moore-Penrose solution. The constrained case is then examined for both a direct and an indirect solution technique: the direct method is a modified version of Hildreth's quadratic programming procedure, and the indirect method is the Extended Moore-Penrose Technique. A four-link arm example serves to illustrate these approaches for avoiding singularities while minimizing joint velocities along a straight line end-effector path.

2 REDUNDANCY PROBLEM

The redundancy problem can be formulated as follows. Let $\underline{x}$ be a vector of dimension m representing the desired positions of the end-effector in workspace coordinates, and let $\underline{\theta}$ be a vector of dimension n representing the joint positions of the manipulator. Typically, the θ_i represent joint angles, but they many also represent distances as in the case of a prismatic joint. If n>m, the manipulator is "kinematically redundant".

The constraint that the end-effector must follow a particular trajectory through the workspace can be expressed by the set of m equations:

$$\underline{x} = \underline{f}(\underline{\theta}) \tag{1}$$

This constraint can also be expressed as a velocity constraint by differentiating (1):

$$\dot{\underline{x}} = J\,\dot{\underline{\theta}} \tag{2}$$

where J, the partial of $\underline{f}$ with respect to $\underline{\theta}$, is called the "Jacobian".

A trivial way of resolving the redundancy is to simply choose a set of n-m constraints $\underline{\alpha}(\underline{\theta})$. This approach is often referred to as the "Extended Jacobian Technique" because the additional constraints can be treated in a manner exactly analogous to $\underline{x}$ in equation (1). Thus the derivative of

$\underline{\alpha}$ can be rewritten as

$$\dot{\underline{\alpha}} = A\,\dot{\underline{\theta}} \tag{3}$$

where A is the partial of $\underline{\alpha}$ with respect to $\underline{\theta}$. Combining (2) and (3) yields

$$\begin{bmatrix} \dot{\underline{x}} \\ \dot{\underline{\alpha}} \end{bmatrix} = \begin{bmatrix} J \\ A \end{bmatrix} \dot{\underline{\theta}} \tag{4}$$

The coefficient of $\dot{\underline{\theta}}$ in (4) is the called the"extended Jacobian" because it is obtained by adding rows to the original Jacobian to account for the $\underline{\alpha}$ constraints. Since the extended Jacobian, J_{ext}, is square, $\dot{\underline{\theta}}$ can be determined directly from (4) by inverting J_{ext} and multiplying by the end-effector/constraint velocity vector.

Though simple in concept, the extended Jacobian approach is fraught with problems unless the additional constraints are chosen carefully. If the additional constraint has a simple geometric interpretation, then the approach may be viable. Otherwise, it is very difficult to show analytically that the extra constraint can always be achieved given the end-effector path to be followed. Walker and Marcus [2] present successful results for composite constraints which realize singularity avoidance and obstacle evasion, but it is not clear whether their constraint is reachable throughout the workspace. Another possible side-effect of introducing nonphysical constraints is the inducement of large joint velocities due to complex joint interactions. In addition, the constraint generally does not realize a performance objective.

One nice feature about this approach is that cyclic motion is attained through the invertible mapping between the extended end-effector coordinates and joint coordinates. If the subset of the workspace in which paths can be tracked is simply connected, Baker and Wampler [3] have shown that closed paths in the workspace produce closed paths in the joint space. Thus when the end effector returns to its starting position, the arm linkages are also in their initial configuration. This will prevent the manipulator from drifting when the end-effector is performing repetitive tasks involving closed paths. This feature is not unique, however, as position-based optimizations also exhibit this behavior.

3 TRAJECTORY OPTIMIZATION

Instead of resolving the n-m excess degrees of freedom by adding constraints, the optimization approach takes advantage of the excess degrees of freedom by maximizing a performance objective. The computational complexity will depend upon the cost used and how constraints are incorporated in the optimization. Since the effect of the contraint is only strong near a constraint boundary, undesirable side effects in the arm's behavior are often averted. In addition, extra singularities in the generalized inverse are introduced only when the inequality constraints are active.

The presence of inequality constraints and computational considerations often lead researchers to choose local over global optimization procedures. Sometimes minimizing a local cost may yield better performance than minimizing a global cost. For example, it may be better to minimize the instantaneous velocity rather than the integral if one is interested in limiting the kinetic energy present in the system.

A local optimization involves minimizing an instantaneous or "local" cost. One cost function which has received a lot of attention is the velocity norm

$$C(\dot{\underline{\theta}}) = \frac{1}{2}\,\dot{\underline{\theta}}^T S\,\dot{\underline{\theta}} \tag{5}$$

Often the weighting matrix S is chosen to be the n x n identity matrix so that one is seeking the minimum norm solution for $\dot{\underline{\theta}}$. Adjoining the velocity constraint (2) to the cost (5) yields the Hamiltonian

$$H(\dot{\underline{\theta}}) = C(\dot{\underline{\theta}}) + \underline{\lambda}^T (\dot{\underline{x}} - J\dot{\underline{\theta}}) \tag{6}$$

Minimizing H with respect to $\dot{\underline{\theta}}$ and $\underline{\lambda}$, and solving the resulting equations for $\dot{\underline{\theta}}$ yields

$$\dot{\underline{\theta}} = S^{-1}J^T (J\,S^{-1}J^T)^{-1}\,\dot{\underline{x}} \tag{7}$$

The coefficient matrix of $\dot{\underline{x}}$ is the weighted right pseudoinverse, J_s^{-1}, of the Jacobian since $JJ_s^{-1}=I$. The solution in (7) is usually referred to as the "Moore-Penrose Technique" but is sometimes referred to as "resolved motion rate control" [4].

The advantages of the velocity-based algorithm include minimization of joint velocities or kinetic energy (if S is the inertia tensor), and zero joint rates when the end-effector velocity is zero. A natural byproduct of the first property is that the arm is repelled by most singularities because of their associated high joint rates. The latter property is a consequence of the minimum norm

solution for the joint rate which only contains components that contribute to the end-effector velocity. A disadvantage of the Moore-Penrose solution is that it does not have the cyclic property of position-based solutions. The "Extended Moore-Penrose Technique" (to be discussed later) is a means of limiting arm drift in performing repetitive tasks but also does not result in cyclic motion.

The requirement that the manipulator not get too close to a singularity is graphically depicted in Fig. 1 and can be expressed as the union of two inequality constraints

$$\alpha \le \alpha_{min} \cup \alpha \ge \alpha_{max} \tag{8}$$

where α_{min} and α_{max} represent upper and lower bounds on α, respectively. Since the minimization is to be performed over the joint velocity not position, the singularity position constraint in (8) is converted to a velocity constraint

$$\begin{aligned} \dot{\alpha} &\ge 0 \qquad \text{for} \qquad \alpha = \alpha_{max} \\ \dot{\alpha} &\le 0 \qquad \text{for} \qquad \alpha = \alpha_{min} \end{aligned} \tag{9}$$

which is enforced at every time step. Unfortunately, (9) produces a harsh effect when α hits the constraint boundary, and a smoother transition is achieved through the constraint

$$\begin{aligned} \dot{\alpha} &\ge b\,(\alpha_{max}-\alpha) \qquad \text{for} \qquad \alpha > (\alpha_{max}+\alpha_{min})/2 \\ \dot{\alpha} &\le b\,(\alpha_{min}-\alpha) \qquad \text{for} \qquad \alpha < (\alpha_{max}+\alpha_{min})/2 \end{aligned} \tag{10}$$

where $b>0$ is a design parameter which specifies the maximum rate of change in α for a given change in α and is chosen sufficiently large so that (10) only becomes active near the border.

The local minimization problem with equality constraints for the end-effector and inequality constraints for the singularity avoidance can now be formulated from (5), (2) and (10):

$$\begin{aligned} \min_{\dot{\underline{\theta}}} \quad & C = \frac{1}{2}\,\dot{\underline{\theta}}^T S(\underline{\theta})\,\dot{\underline{\theta}} \\ \text{subject to} \quad & J(\underline{\theta})\,\dot{\underline{\theta}} = \dot{\underline{x}} \\ & A(\underline{\theta})\,\dot{\underline{\theta}} \le \underline{c} \end{aligned} \tag{11}$$

where A is the gradient of α with respect to $\underline{\theta}$ and $\underline{c}$ is the right side of the inequality constraint in (10). (A negative sign precedes these quantities in A and $\underline{c}$ for the "$\ge$" constraints.) Hildreth's solution [5] to the above problem modified to account for the equality constraints [6] is given by

$$\dot{\underline{\theta}} = -\,S^{-1}\,Q^T\,\underline{\nu} \tag{12}$$

$$\nu_i^{k+1} = \begin{cases} w_i^{k+1} & i = 1 \ldots m \\ \max\,(0\,,\,w_i^{k+1}) & i = m+1 \ldots m+p \end{cases}$$

$$\text{where} \qquad w_i^{k+1} = -\frac{1}{P_{ii}}\left[d_i + \sum_{j=1}^{i-1} P_{ij}\nu_j^{k+1} + \sum_{j=i+1}^{m+p} P_{ij}\nu_j^{k} \right]$$

$$\text{and} \qquad P \equiv Q\,S^{-1}Q^T\,,\ \underline{\nu} = \begin{bmatrix} \underline{\nu}_E \\ \underline{\nu}_I \end{bmatrix},\ \underline{d} = \begin{bmatrix} \dot{\underline{x}} \\ \underline{c} \end{bmatrix},\ Q = \begin{bmatrix} J \\ A \end{bmatrix}$$

where $\underline{\nu}_E$ are the Lagrange multipliers for the end-effector equality constraints in (11), $\underline{\nu}_I$ are the multipliers for the p inequality constraints, and k is the iteration number.

An alternative way of including position inequality constraints is the Extended Moore-Penrose Technique. Unlike the direct Lagrangian approach just developed, the position constraint is realized indirectly by including a position term in the unconstrained Moore-Penrose solution which forces a gradient descent away from the obstacle

$$\dot{\underline{\theta}} = J_s^{-1}\dot{\underline{x}} + \rho\,(I - J_s^{-1}J)\,\nabla P(\underline{\theta}) \tag{13}$$

where ρ is a positive scalar constant, and $P(\underline{\theta})$ is an objective function to be maximized. The first term in (13) is the minimum norm solution for $\dot{\underline{\theta}}$ which is the unconstrained solution to (5). The second term is in the nullspace of J and thus contributes nothing to the end-effector motion. Thus ∇P can have any value and still not perturb the constraint in (2).

Liegeois [7] experiments with objective functions for a six joint revolute manipulator to minimize joint deviations while tracking a circle with the end-effector. Klein and Huang [8] report results for a similar cost using a three-link manipulator being commanded by a joystick. Baillieul et al [9] consider a manipulability criterion for a planar four-link mechanism and demonstrate the ability of the extended method to keep the manipulator away from the singular initial configuration. Only Klein and Huang discuss the effects of varying the size of ρ, emphasizing the need to scale it for different end-effector positions.

4 CONSTRAINED VELOCITY OPTIMIZATION EXAMPLE

A schematic of the manipulator used in our study is shown in Fig. 2. The K1607 Robotics Research® Arm has seven revolute joints in an alternating roll/pitch sequence beginning with the shoulder roll at the base. The Denavit-Hartenberg (D-H) parameters for the first five joints of this arm are given in Table I. In this study, we will only be concerned with controlling the cartesian position of the wrist, the origin of frame 5 in Fig. 2 which is only a function of the first four joint angles.

Table I: D-H Parameters for Robotics Research K1607 Manipulator.

joint # i	a_i (cm)	α_i (deg)	d_i (cm)
1	0.00	-90	0.00
2	14.29	+90	0.00
3	10.80	-90	68.58
4	-10.80	+90	0.00
5	-	-	68.58

In the simulation examples, the trajectory planner will be attempting to avoid the singularities formed by the intersection of the two planes [6]

$$\frac{\sin(\theta_2)}{\cos(\theta_2)} = \frac{-a_2 f_2(\theta_4)}{b_3 f_2(\theta_4) - a_3 f_1(\theta_4)} \quad \text{and} \quad \cos(\theta_3) = \frac{a_2 f_1(\theta_4)}{b_3 f_2(\theta_4) - a_3 f_1(\theta_4)} \tag{14}$$

where $f_1(\theta_4) \equiv d_5 c_4 + a_3 s_4$ $\quad c_i \equiv \cos(\theta_i)$

$f_2(\theta_4) \equiv d_5 s_4 - a_3 c_4$ $\quad s_i \equiv \sin(\theta_i)$

These conditions can be used to form a manipulability measure which goes to zero at the line singularity:

$$\alpha(\underline{\theta}) = |f_a| + |f_b| \tag{15}$$

where $f_a \equiv c_3(d_3 f_2 - a_3 f_1) - a_2 f_1$

$f_b \equiv s_2(d_3 f_2 - a_3 f_1) + c_2 a_2 f_2$

In the direct Lagrangian method, α was used for the inequality constraint in the problem formulation (11). The upper and lower bounds on α were 0.20 m and -0.20 m, respectively, and the slope parameter, b, was 10.0 s^{-1}. Because the Lagrange multiplier solution is numerical, a sufficiently close approximation may take several iterations. In the extended Moore-Penrose method, α was used for P, the position objective. The scalar weighting for the nullspace component, ρ, was chosen to be 0.15 which yielded a minimum manipulability of about 0.2 m (the same as the direct Lagrangian approach). In both cases, the unweighted velocity norm was used for the objective function.

The end-effector was commanded to follow a straight-line path in the workspace from an initial joint configuration of $\underline{\theta}^T$ = [-0.1 -1.0 0.8 1.3] rad which corresponds to $\underline{x}^T$ = [-0.40 0.68 1.08] m to a final position of $\underline{x}^T$= [0.6 0.7 0.5] m. The path acceleration was constant to the midpoint followed by a constant deceleration of the same magnitude to the end point.

Ten or more iterations on the Lagrange multipliers were often needed before the Lagrange algorithm converged. The primary driving force behind the convergence rate was the size of the Lagrange multiplier for the inequality constraint relative to the end-effector multipliers. The larger the constraint violation in the unconstrained case, the larger the multiplier and the slower the convergence. Thus the drastic change in the multiplier for the first few iterations propagated through the other multipliers and caused errors in the end-effector velocities as well.

The degree to which the unconstrained (Moore-Penrose) solution was affected by the manipulability constraint in the Lagrangian approach is seen in Fig. 3. The joint velocities were unconstrained until about 1.3 s when the manipulator got close to the line singularity. At this point, the manipulability constraint took effect preventing further acceleration toward the preset threshold at 0.2 m. Before the manipulability reached this boundary, the unconstrained solution no longer violated the constraint, and the solution reverted back to Moore-Penrose. The curves do not coincide exactly from this point because the joint configurations were altered by the constraint.

The shoulder and elbow joint velocities (joints 2 and 4) in the Lagrangian-constrained (converged) and unconstrained cases are shown in Fig. 4. These curves more dramatically illustrate

the effect of the manipulability constraint on the arm. All of the unconstrained velocities peaked when the manipulability reached the minimum in Fig. 3. When the constraint went active, joint 2 underwent a discontinuous jump in acceleration resulting in a decrease in velocity. The constraint caused a significant but less abrupt decrease in speed for joint 4. Joints 1 and 3 (not shown) underwent behavior paralleling that of joints 2 and 4, respectively.

The Extended Moore-Penrose approach displayed smoother behavior than the Lagrangian method. Since the nullspace component was being continuously applied rather than only when an inequality constraint was being violated, the manipulability measure for the Extended Moore-Penrose solution in Fig. 5 always differed from that for the unconstrained solution (ρ=0). At the beginning of the trajectory, the manipulabilities did not differ greatly, but as the arm drew closer to the singular configuration, the nullspace component became dominant, driving the manipulator away from the singularity. When tracking the end-effector trajectory placed the singularity out of reach, the Extended Moore-Penrose solution once again approached the Moore-Penrose solution.

The pitch joint velocities in the Extended Moore-Penrose case are plotted along with the unconstrained results in Fig. 6. The velocity of joint 4 was remarkably similar to that for the Lagrangian approach, but the velocity for joint 2 did not make the same sharp transition near the singularity. This is because the activity of the inequality constraint in the Lagrangian solution is binary, and thus does not possess the continuous properties of the Extended Moore-Penrose approach. Thus, whilst the constant application of the nullspace component is less efficient, it does yield the advantage of continuous joint velocities.

The cycle computation times for the Extended Moore-Penrose and Lagrangian Techniques were 2.8 msec and 2.9+0.3k msec, respectively, where k is the number of Lagrange multiplier iterations. By comparison, the unconstrained solution took 2.2 msec, and the extended Jacobian method took 2.0 msec. These times are for Microsoft® C 5.1 running on an Intel® 80386/7 and do not include the 1.4 msec it takes to compute trigonometric functions of the joint angles.

Although the Extended Jacobian Method would provide the fastest solution, the measure in (15) would not be a good choice for the constraint because of its limited range. For example, attempting to hold α constant for the same initial configurations used above fails at 1.3 sec because it cannot satisfy the specified end-effector trajectory. Kreutz-Delgado et al [10] suggest using an elbow roll angle for α in their work with a similar arm, but this further restricts the workspace and does not avoid singularities without knowing a priori which α's cause singularities.

5 CONCLUSIONS

It was discovered that position constraints can be included in the optimization either directly or indirectly. For a quadratic cost and linear constraints, the problem can be solved via Hildreth's quadratic programming algorithm. This approach incorporates the inequality constraints directly by iterating on the Lagrange multipliers. A second approach, the Extended Moore-Penrose method, realizes the constraint by adding a gradient component to the solution which steers the arm away from the constrained region.

These two approaches were examined for the case of a four-link arm operating in 3-D space. The goal for this example was to prevent the arm from entering a region surrounding a line singularity while following a straight line path with the end-effector. The Lagrangian solution avoided the singularities but had sharp velocity transitions. The Extended Moore-Penrose solution smoothly avoided the singularity but at the expense of optimality in the velocity norm. The Extended Moore-Penrose method was more than twice as fast as a ten-iteration Lagrangian when a single constraint was being applied.

The extension of optimization-based algorithms to include inequality constraints makes them a more attractive option for redundancy resolution. Instead of simply adding more constraints to reduce the excess degrees of freedom to zero, the designer can now utilize the redundancy to his advantage by maximizing some performance objective critical to the task while avoiding undesirable regions of the workspace. In addition, the penalty for including these constraints may be less than 50% of the total trajectory computation time for the extended Jacobian solution. Coupled with new results on their global characteristics, local optimization methods have now become a viable contender for real-time trajectory planning.

ACKNOWLEDGEMENTS

The authors would like to express their appreciation to the Space Station Project Office (Code 400.6), NASA Goddard Space Flight Center for their support of this research.

REFERENCES

1. Baillieul, J., "Kinematic Programming Alternatives for Redundant Manipulators," Proc. IEEE Int. Conf. on Robotics and Automation, St. Louis, March 25-28, 1985, pp. 722-728
2. Walker, I., and Marcus, S., "Subtask Performance by Redundancy Resolution for Redundant Robot Manipulators," IEEE J. Robotics and Automation, Vol. 4, No. 3, June 1988
3. Baker, D., and Wampler, C., "On the Inverse Kinematics of Redundant Manipulators," Int. J. Robotics Research, Vol.7, No. 2, Apr. 1988
4. Whitney, D., "Resolved Motion Rate Control of Manipulators and Human Prostheses," IEEE Trans. on Man-Machine Systems, Vol. MMS-10, No. 2, June 1969, pp. 47-53
5. Luenberger, D., Optimization by Vector Space Methods, Wiley & Sons, New York, ©1969, pp. 299-300
6. Carignan, C., "Trajectory Optimization for Kinematically Redundant Arms," Tech. Rep. No. STX/ROB/90-01, STX Corp., Lanham, Md., January 1990
7. Liegeois, A., "Automatic Supervisory Control of the Configuration and Behavior of Multibody Mechanisms," IEEE Trans. on Systems, Man, and Cybernetics, Vol. 7, No. 12, 1977
8. Klein, C., and Huang, C.-H., "Review of Pseudoinverse Control for Use with Kinematically Redundant Manipulators," IEEE Trans. on Systems, Man, and Cybernetics, Vol. SMC-13, No. 3, March 1983, pp.245-250
9. Baillieul, J., Hollerbach, J., and Brockett, R., "Programming and Control of Kinematically Redundant Manipulators," Proc. of 23rd Conf. on Decision and Control, Las Vegas, Dec. 1984, pp. 768-774
10. Kreutz, K., Long, M., and Seraji, H., "Kinematic Analysis of 7 DOF Manipulators," JPL, Pasadena, (in progress)

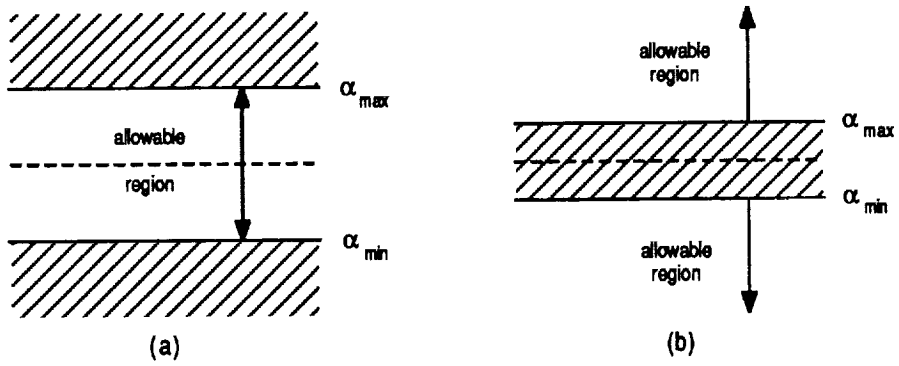


Figure 1: Allowable regions for (a) joint limit and (b) singularity constraint types.

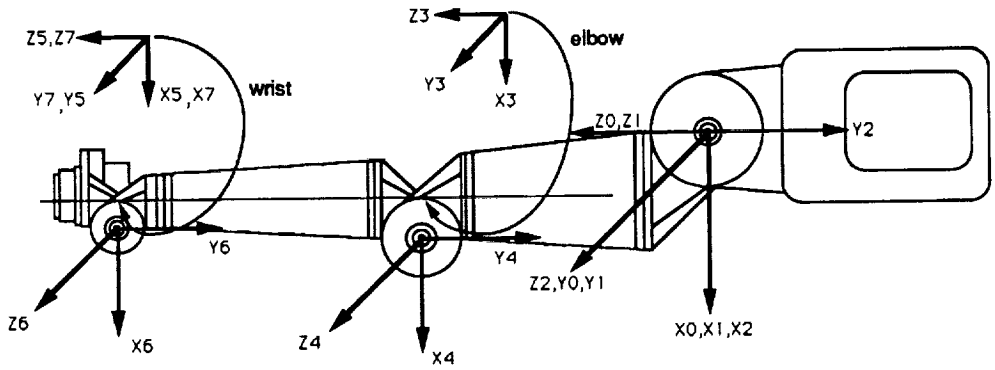


Figure 2: Robotics Research K1607 Manipulator used in simulations.

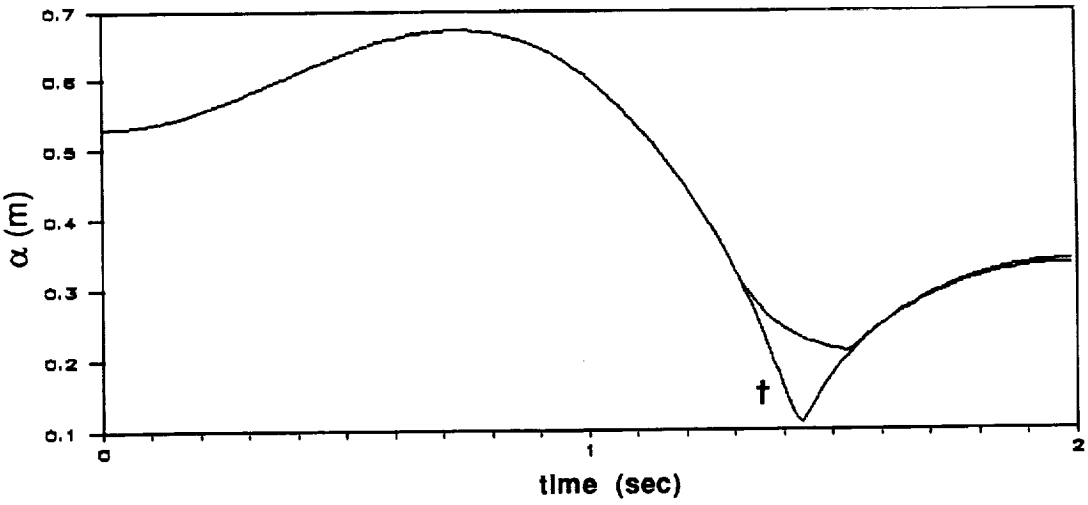
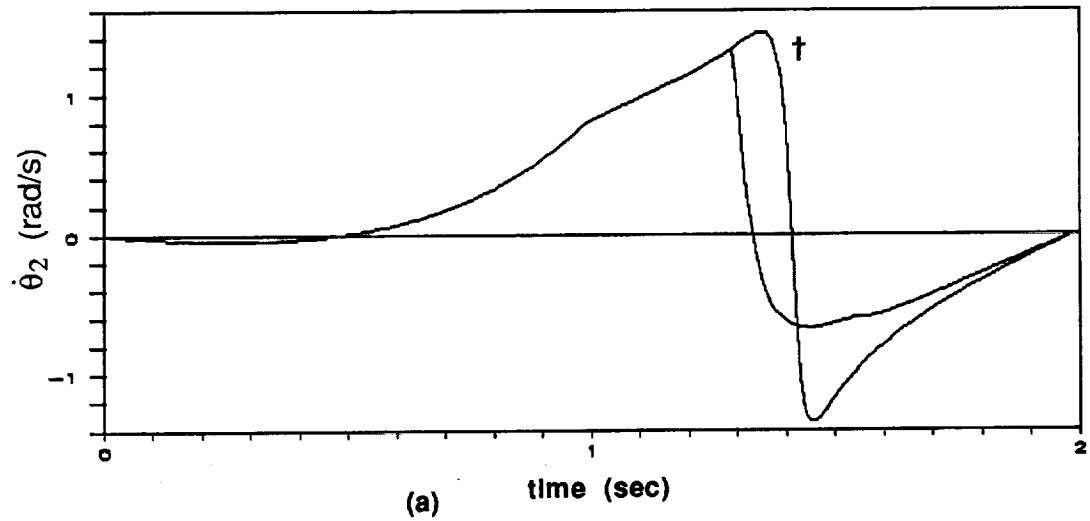
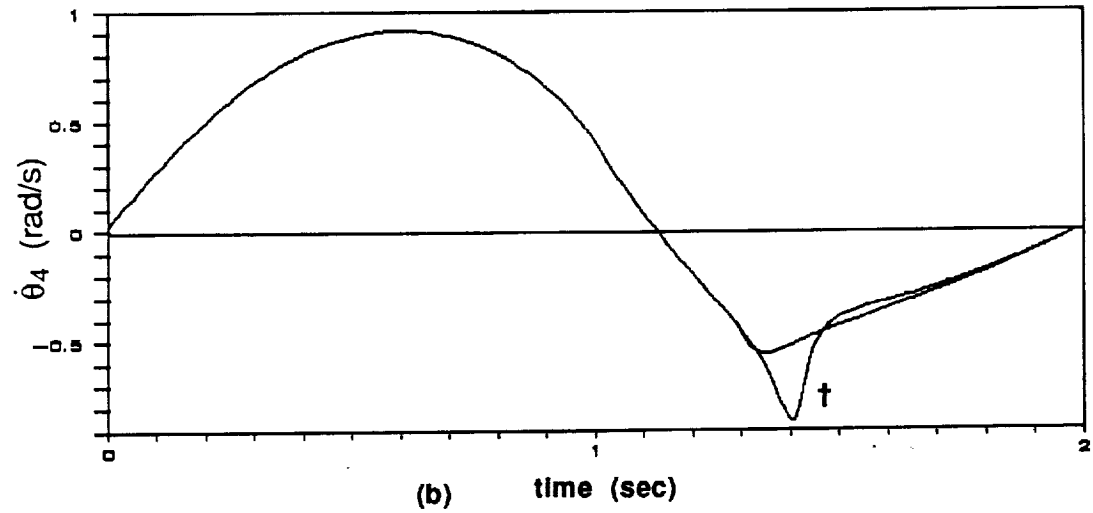


Figure 3: Manipulability for the constrained Lagrangian case is plotted versus the unconstrained case (†).



(a)



(b)

Figure 4: (a) Shoulder and (b) elbow pitch velocities for constrained Lagrangian case versus unconstrained (†).

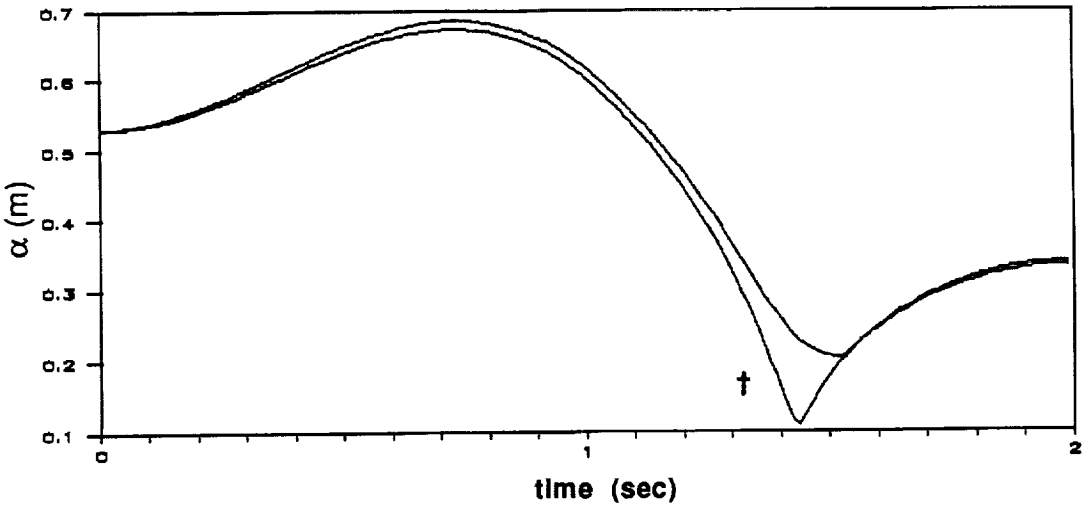
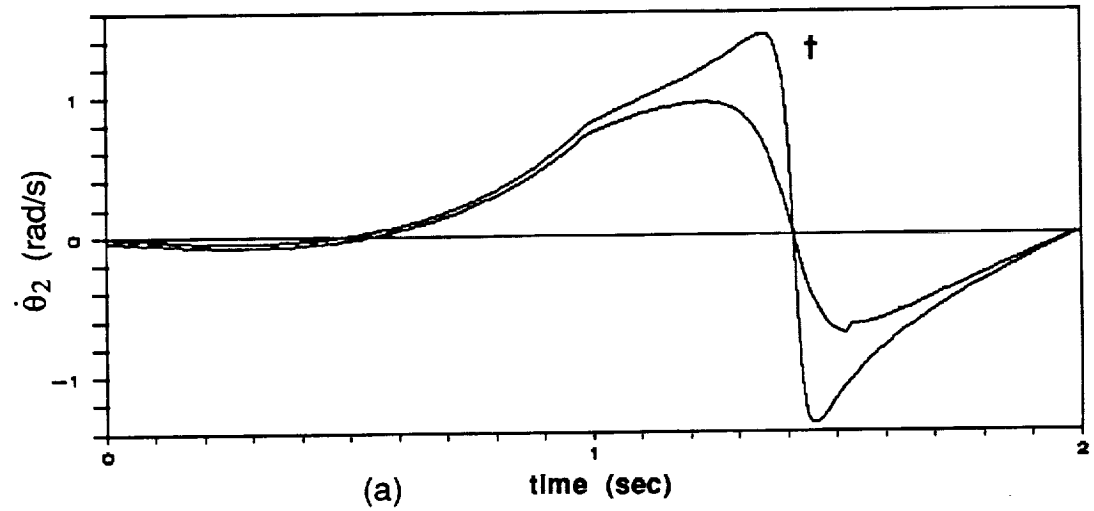


Figure 5: Manipulability for the constrained Moore-Penrose case is plotted versus the unconstrained case (†).



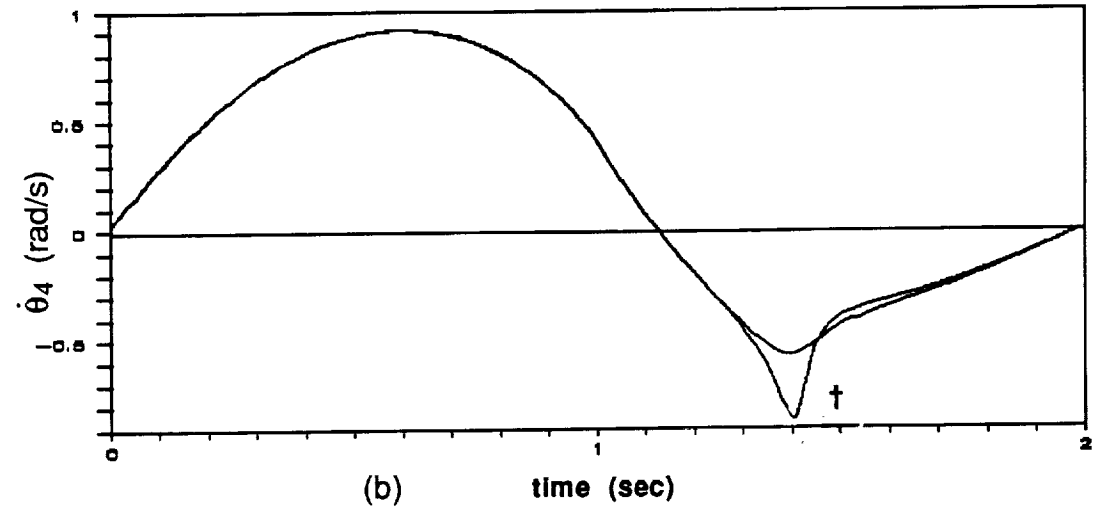


Figure 6: (a) Shoulder and (b) elbow pitch velocities for constrained Moore-Penrose case versus unconstrained (†).